

A Cybernetic Approach to the Modeling of Agent Communities

Walt Truskowski, Senior Technologist, Code 588
NASA - Goddard Space Flight Center, Greenbelt, MD 20771
Walt.Truskowski@gsfc.nasa.gov

Jay Karlin, Viable Systems Inc
12236 Stoney Bottom Road, Germantown, MD 20874
jkarlin1@ix.netcom.com

Abstract

This paper, which is expository in nature, investigates and illustrates the idea of using cybernetic system modeling techniques to model a multi-agent community. The cybernetic concepts used are those originally developed by Stafford Beer. The paper illustrates the idea on a community of agents being developed at Goddard to support both ground-based and space-based system autonomy. It is hoped that use of this approach will help provide a deeper insight into the dynamics of agent-community behaviors. Time and additional analyses will tell.

Keywords: Agents, Agent Communities, Autonomy, Autonomous Systems, Cybernetic System Modeling, Agent Communication Language, Artificial Intelligence, Human-to-Computer

A Cybernetic Approach to the Modeling of Agent Communities

Walt Truszkowski, Senior Technologist, Code 588
NASA - Goddard Space Flight Center, Greenbelt, MD 20771
Walt.Truszkowski@gsfc.nasa.gov

Jay Karlin, Viable Systems Inc
12236 Stoney Bottom Road, Germantown, MD 20874
jkarlin1@ix.netcom.com

Abstract

This paper, which is expository in nature, investigates and illustrates the idea of using cybernetic system modeling techniques to model a multi-agent community. The cybernetic concepts used are those originally developed by Stafford Beer. The paper illustrates the idea on a community of agents being developed at Goddard to support both ground-based and space-based system autonomy. It is hoped that use of this approach will help provide a deeper insight into the dynamics of agent-community behaviors. Time and additional analyses will tell.

1. Introduction

In an earlier paper [1] examples of agent technology in a NASA context were presented. Both ground-based and space-based applications were addressed. This paper continues the discussion of one aspect of the Goddard Space Flight Center's (GSFC) continuing efforts to develop a community of agents that can support both ground-based and space-based systems autonomy. The paper focuses on an approach to agent-community modeling based on the theory of viable systems developed by Stafford Beer. It gives the status of an initial attempt to capture some of the agent-community behaviors in a viable system context. This paper is expository in nature and focuses on a discussion of the modeling of some of the underlying concepts and infrastructure that will serve as the basis of more detailed investigative work into the behavior of agent communities. The paper is organized as follows. First, a general introduction to agent community requirements is presented. Secondly, a brief introduction to the cybernetic concept of a viable system is given. This concept forms the foundation of the modeling approach. Then the concept of an agent community is modeled in the cybernetic context.

2. Agent Communities in General - Requirements

In this particular paper we are not specifying a particular agent architecture. We are, however, assuming that the agent has the capability for reactive, deliberative, reflexive, and social behaviors. The particular agent architecture that we are using at Goddard is a component-based architecture implemented in

Java which is capable of all four types of behaviors [1]. Our community is populated with agents capable of these behaviors.

We begin by identifying and briefly discussing what are the general requirements for an agent community from our perspective. These requirements serve to establish the general context for understanding agent community concepts. Good sources for agent community concepts can be found in [2., 3]. These are the requirements to be modeled.

1) An agent community will have an overarching goal and shall accommodate subgoals.

This overarching goal establishes a "purpose" and this purpose makes the agent community a "system" in the cybernetic sense.

2) Any two agents in the community can carry on a meaningful conversation. Thus, all agents in the community shall have a shared ontology.

This shared ontology may be the kernel of a larger ontology which is obtained through the "composition" of the ontology's of all of the agents in the community. This kernel is necessary to support ontology negotiation between agents who wish to collaborate but do not have identical ontology's.

3) An agent community shall have at least one user interface.

This interface to the outside world (user) provides a mechanism for allowing the user to establish, in real-time, new goals for the community, to observe the behavior of the community in action and to get status information on the community's activities.

4) An agent community shall have an infrastructure capable of supporting its internal communications.

Without such an infrastructure the agents would exist in isolation from one another and not be able to function as a community.

5) An agent community infrastructure shall be capable of supporting the cooperative behavior of its members.

Cooperative behavior may be broken down into either coordinated or collaborative behaviors. In coordinated behavior a single agent is vested with authority over other agents. In collaborative behavior, the agents that are cooperating do so as peers.

6) An agent community will be adaptable to environmental changes.

Continual environmental changes are to be expected. Adaptability can be realized in several ways including intervention from the "user" and agent learning (however realized).

7) Agent community members will negotiate and share agreements.

This is the essence of collaborative behavior in a community.

8) An agent community will have integrated reasoning ability.

This means that a group of agents will have the capability to collectively reason about a problem utilizing the talents, knowledge, etc. of all the agents in the group. Another way of saying this is that the community is a knowledge community.

9) *An agent community shall reason about at least one domain.*

This is related to Requirement 1 above. The goal of the community is to serve with respect to, at least, one domain of activity.

10) *An agent community will be capable of negotiating their shared interests and individual goal priorities in several different subdomains.*

This is related to both Requirements 1 and 9. As an example: if the domain of activity is spacecraft operations then the community as a whole is responsible for monitoring and maintaining successful spacecraft operations. An agent may be associated with the power subsystem, another with the thermal subsystem, yet another with command and control. These agents with their own subdomain interests may work together to ensure a broader domain interest.

11) *An agent community shall have access to plans or partially complete plans.*

We view a community as a knowledge community and as such the knowledge level of the community rises with each individual agent success. The plans that were used to successfully accomplish a task become part of the knowledge base of the community for future use by the community. The community knowledge base may initially (when the community comes into existence) be empty.

12) *An agent community will have a history-keeping and logging capability.*

This is part of the documentation of the community's knowledge base. In order for the community to both improve its performance over time as a community and to be responsive to "outside" queries about its behavior patterns a logging mechanism is required.

3. Viable Systems - a Cybernetic View

A system is defined as a combination of components, which interact in order to perform an identifiable service or set of services. An environment that receives these services and also may, in turn, alter the system in some way surrounds the system. Such an arrangement is shown in Figure 1. For example (in a spacecraft context), conditions resulting from thermal or other environmental or internal system effects can reduce a system's life. We define a system's viability as its functional persistence. A viable system is a robust one: it adapts its own behavior mode, structure, etc., to provide its services even under duress. An intelligent agent assigned in the system may monitor and regulate health or even direct system performance. System health and performance would then become the agent's domain.

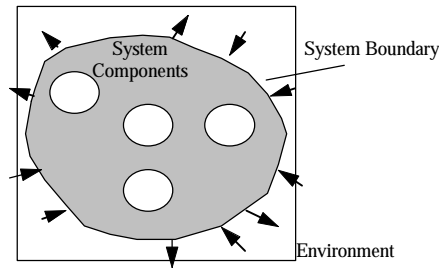


Figure 1. System Defined

A system performs services as tasks in order to effect changes in its environment. The system strives to meet goals which have been decided upon from some higher commanding order of intelligence, e.g., a user. Intelligent agents may be considered for inclusion in this higher order if they are able to act as community management on the behalf of a user to achieve a desired environmental state. This state-oriented agent could be capable of judgment calls or of convening a meeting of agents collaborating on a strategy which would then be parsed out for execution to, for instance, specialist/tactical agents.

The viable system architecture provides a way of discussing internal and external system behaviors in a systematic manner. It involves 5 levels of recursion in doing so. The viable system model depicted in Figure 2 has both horizontal and vertical recursive paths. This model was originally developed by Stafford Beer [4,5,6,7,8,9] as part of his management cybernetic work. We are attempting to utilize the concepts and notations to gain comprehensive insight into the needed behaviors of agent communities.

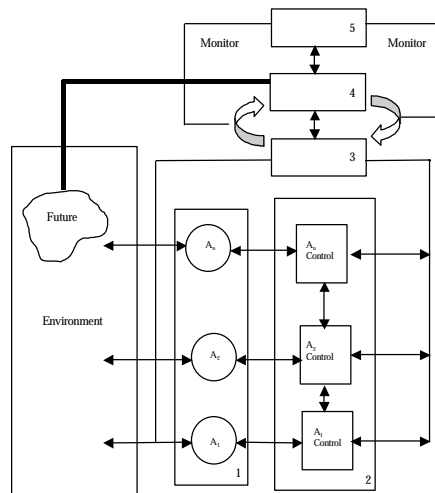


Figure 2. Viable System Model Schematic

The model consists of 5 systems, numbered 1-5. System 1 is the base system, System 2 provides local regulatory monitoring and control for system 1. It regulates oscillatory behaviors. System 3 deals with self-organization and autonomic regulation. System 4 deals with self-reference, simulation and planning. System 5 deals with homeostasis and overall policy making.

In preparation for the rest of the paper let's begin with a preliminary application of this model schematic to an agent community. The overall system, which is being modeled, is the community of agents. System viability refers to the fact that the community can withstand perturbations either triggered by the environment or by internal changes in individual agent's behaviors. The various systems 1-5 identified in the schematic can be thought of as encapsulations of entities and/or functionality that contribute to the overall viability of the agent community. The elements of System 1 will be individual agents A_i . These agents interact with the environment through perceptors and effectors to maintain an awareness of the environment and to make impacts on the environment. System 2 is concerned with being aware of the behaviors of the elements in System 1 and providing behavioral control. System 3 focuses on the organization of the community and the autonomy that the elements in the community have to contribute to self-organization. For example, the formation of a subgroup of agents to focus on a specific problem would come under the purview of the functionality in System 3. System 3 is concerned with things as they are. System 4 is concerned with what is going on in the environment and what needs to be done to prepare for the future. System 4 contains functionality that enables itself to maintain a view of itself as a community. There are many interactions between Systems 3 and 4 as is depicted in the schematic. System 5 monitors the interaction between Systems 3 and 4 and establishes overall policies for the agent community as a whole.

4. Agent Communities from a Cybernetic Perspective

We have briefly introduced the concept of a viable system framework. We now delve deeper into the utilization of this approach in the modeling of agent communities. The examples used come from an agent community whose domain is a spacecraft.

4.1 Intelligent Agent Framework for Communities:

To say an agent is intelligent is to imply, among other things, the existence of what we call adaptive functionality. In general, adaptively functional agents are capable of doing three things:

- Noticing: trying to detect potentially relevant events from the environment.
- Interpreting: trying to recognize the events (generally this means mapping the external event into an element in the domain system's

vocabulary, i.e., a model) by applying a set of recognition rules.

- Responding: acting on the interpreted events by using a set of action rules, either by taking some action that affects the environment , or by altering their own rules (i.e., learning).

It is the authors' opinion that one of the main differences between intelligent agent systems and other Artificial Intelligent, (AI), systems is a matter of their relative degrees of independence. For example, an agent can decide that in order to complete its task it must look outside to find required expertise or knowledge. It can then proceed to search for and use such a resource in order to complete its task. In contrast, under similar circumstances, the AI system would issue a message to the user that it could not complete the assignment (e.g., within its constraints) and then go into standby or await additional instruction. This ability of an agent, to reflect upon its own capabilities and then to actualize itself, either in commencing a learning routine or engaging another agent and collaborating with it, is what distinguishes intelligent agents from other types of autonomous systems. This capacity would be preferred over more limited AI capabilities in applications where, for instance, an agent is inaccessible to direct reprogramming, such as in missions with limited ground station coverage, but has access to other agents nearby.

4.2 Agent Communities Parallelism

The community shown in Figure 3 is the embodiment of the higher order system introduced in the previous paragraph. This community, in order to maintain its viability whenever its domain undergoes stress, facilitates its member agents in responding appropriately.

The white areas, in Figure 3, surrounding each of the three systems depict environment or domain areas. Agent communities are composed of specialists and facilitators or “system managers”.. Architectural types may be hierarchical, distributed, distributed hierarchical or clustered, depending on the goal of the application. In this paper we intend to describe their possibilities and model some useful structural-behavioral characteristics. Note that the community concept makes no sense with an AI system, but it is here in situations involving homeostasis that autonomous agents realize their fullest potential.

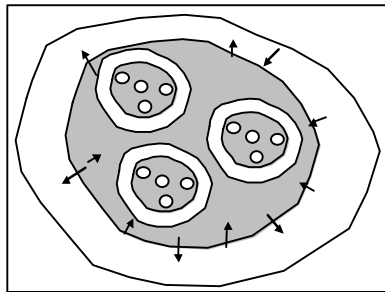


Figure 3. Three agents comprising a community, with their associated domains embedded within the community's domain. Domains are represented by encircling white bands.

The members cooperatively adapt with new plans and initiatives to meet the challenge presented in order to exercise their community responsibility. These new measures are assigned to tactical agents to minister to their individual domains. The procedure unfolds differently according to the following cases:

1. Community-level perception/decision resulting in new goal assignments (possibly new models) to the agents.
2. Attempts to reconcile, with existing models, different simultaneous anomalies detected by two or more agents.
3. Community intervention within an agent's domain should a crisis situation demand it.
4. Community learning by formulating revised models of its domain.

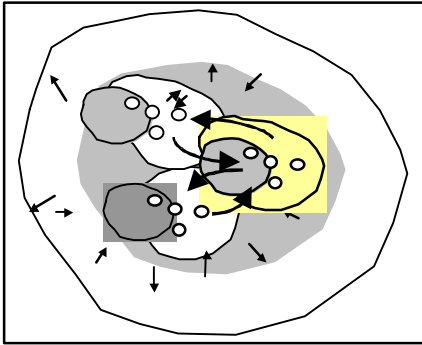


Figure 4. Intersection of Domain Areas Calls for Domain Related Adaptive and Cooperative Behavior by their Respective Agents

4.3 Agent Collaboration

In addition to community-agent interaction there is agent-agent interaction. A community's domain is normally made up of sub-domains, which overlap one another because of the interdependence of subsystems. This interdependence is depicted in Figure 4. by the overlap of the agents' domain areas as these areas undergo change. Since a single agent pays attention to only a subset of inputs (percepts), cooperation is required in order to provide more comprehensive coverage. Case 2 above is one example where this applies.

Therefore, in order to examine cooperation in an agent community it is helpful to re-map the previous figures to clarify agent and system interactions. A graphic demonstration follows in which subsystems are shredded out of their environments and domain-associated agents shredded out of their systems. For a single agent the result appears as shown in Figure 5A. Figure 5B shows the interaction of three agents and their respective domains/subsystems.

4.4 Community Model Framework

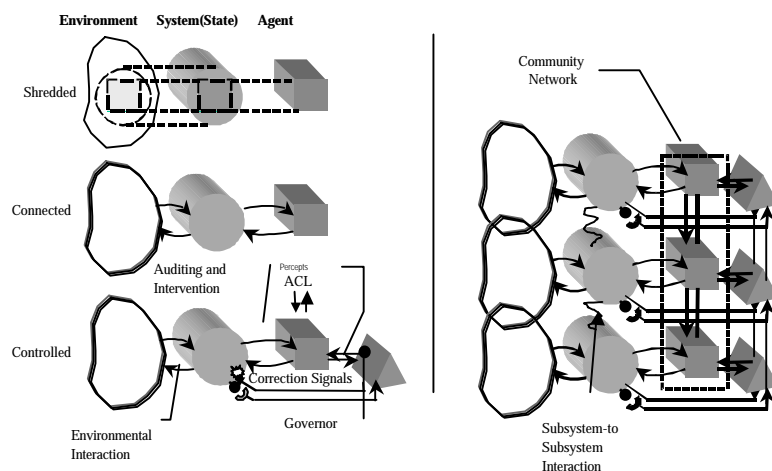


Figure 5A Instantiation of an Agent/Domain and Figure 5B Interaction of Agents as a Community

Conceptually, Figure 5A shows the system and its domain-associated agent having two interfaces to the system (horizontal arrowheads): one interface is a triangular prism, which represents a reactive correction device or governor. The second is a direct intervention or override auditing function which analyzes spurious movements of the system and attempts to redirect it from a revised perspective or model (i.e., learning). Both interfaces are stimulus/response types but the latter interface is a probing by the agent into different critical nodes of its associated system on an intermittent basis while the former is monitoring and adjusting nominal performance settings. In this way, by use of the probe, the agent can investigate a wider domain but yet limit its own demand for critical community resources. Note that in nominal operation the triangular prism provides a semi-autonomous (i.e., reactive) control to the system. The agent is stimulated to replan only when the governor (triangle) indicates that the system's healthy limits might be violated. Thus the governor forms a part of the agent (its perceptor/effector arrangement) but, as will be shown below, the governor is also a node of the network that coordinates the overall communal system. This discussion begins to illustrate the cybernetic (viable system) modeling technique under investigation.

4.5 Coordination Framework

A communal coordination network is illustrated in Figure 5B as vertical message and data pathways are joining together the set of subsystems' governors and also the set of subsystems' domain-related agents. The former deals with percepts and corrections or effects in system performance and the latter with agent-agent communications via some Agent Communication Language (ACL) messaging. The governors respond to input telemetry signals by changing output values (switch settings or gain controls or the sending of pre-stored commands to system devices or to a command management system). Change messages indicating subsystem mode or other state attributes that could affect adjacent subsystems are communicated to these subsystem agents through this coordination network. When a more informative or complex form of intercommunication is necessary, agent communication language is used to convey such information directly between agents via an ACL message.

In addition to communal coordination there is another type of coordination. This function coordinates relative to communal interests, an example of which is an auditor that enforces resource sharing and corrects transmission of data. For example, if a system begins to draw more than its normal level of electric current, then, even though its performance in terms of system functional performance may be within tolerance, something internal to the system is suspect. For this type of trouble shooting, the special discipline dealing just with such matters needs to be applied. Since a single source normally distributes power to all systems, a central monitor agent would be used having

unique capability to trouble shoot the electrical power system. (There is obviously a tradeoff possibility here) The point is that there may be certain agent and component roles that would be better suited for a central position in the overall schema than as a peer member in the community.

There are a number of these positions that together suggest themselves as a separate central body in the community. Some of them are as follows:

- Agent registrar.
- Planner and Scheduler (from a system shared resource consideration).
- Futures Planner (or “what-if” coordinator).
- Agent translator and communications czar.
- Executive.

Although simpler systems may escape with having only one tier limited to two or three collaborating agents, complex systems may have several tiers of agents, grouped into communities, in which community-to-community communication is carried on. Individuals from two or more communities might in turn occupy additional positions as member of a higher level planning and coordination or executive body.

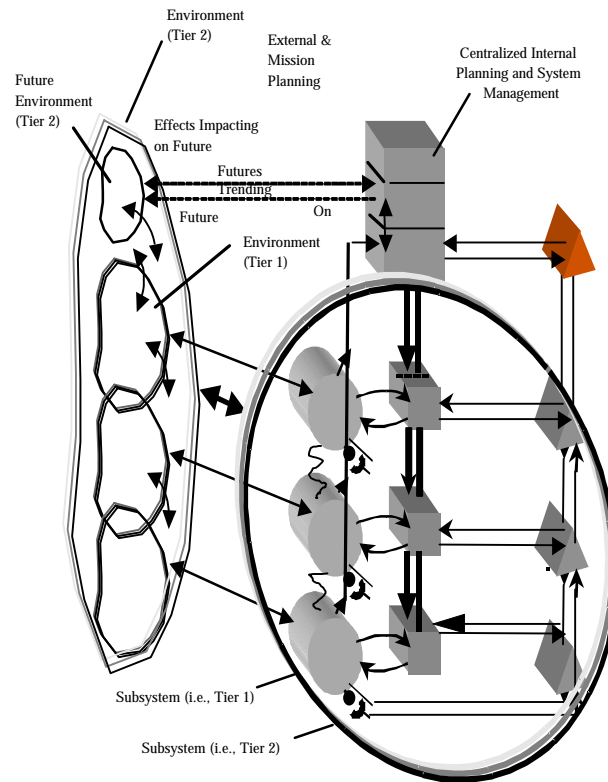


Figure 6. Nested Systems/Agents as Containers

4.6 The Higher Levels of Agent Cognition – Introduction to Centralized Community Planning and Coordination

The central portion of an agent community is shown at the top of the model in Figure 6. It is divided into three segments. The lowest one deals with real-time: events relative to the subsystems as relayed to this real-time segment both from the subordinate agents and through their interconnecting monitoring and coordinating networks. The middle segment provides the system and agent community modeling which provides, in turn, state information for various analytical purposes including plan preparation performed by the real-time segment. The middle segment is also the forward-looking arm of the community in that it is in constant contact with both the user interface (top segment), and with the external environment in order to be able to anticipate events and make advance preparations. The top most segments manage the interface with the user of the agent community. In an agent-governed operational context it provides human-to-computer interaction on an as-needed basis rather than continuously as in the hands-on operator version.

4.7 Recursive Process – System Overall Management Function

In the subject domain, environment changes on the system are stimulators of self-reorganization or adaptation. As a subsystem is altered the information is transmitted through the governor and to the tier 1 agent. Agent action and a reflection of such action is further processed in the coordination network since this governor (or subsystem regulator) is one of that network's nodes in addition to its being a part of the agent-to-subsystem loop. Changes introduced in the agent-to-subsystem loop, if significant enough to affect the overall system, are passed on to the tier 2 management level through the coordination network. Figure 6 illustrates the orthogonality of tiers 1 and 2 in order to illustrate that although the two share instances within the enveloping environment, tier 2 must guard the overall system in such instances, whereas the concern of tier 1 is its narrower domain, e.g., an individual subsystem's health and performance. One might call this effect "information hiding" similarly to structured programming.

If the foregoing discussion has not done so already, the diagram of Figure 6 should have revealed the underlying premise of this report: that the agent community being discussed is an architectural model of the lower part of the central nervous system. If the visceral organs of the body system are represented by the subsystems, then our coordinating and other networks perform the functions of sympathetic and parasympathetic ganglia, while the agent-to agent ACL communication network is the spinal column. And this spinal column is suspended from the cerebellum portion of the brain, which is the lower segment as discussed above. This physiological analogy is based on the work of Stafford Beer [4,5,6].

4.8 Community Builder's Operational Conceptual Model

In contemplating a specific layer or tier, the conceptual model requires consideration of protocols for interaction one tier above and one tier below such

a tier. Different protocols are called for depending on whether the agent's primary function is reactive or deliberative.

4.9 Two Types of Inter-agent Protocols

If we have preprogrammed an agent with a certain low level of capability, then that agent automatically falls into the simple reactive protocol structure. If we, on the other hand, develop an agent capable of high levels of reasoning then the protocol structure needs to fit with the protocol structure that would support the more abstract deliberative forms of information transfer. This would use ACL messaging to signify sender intentions which conveys priority. We propose to place the more abstract protocol ability in the cognitive part of each tier and the simpler or task-oriented protocol in the system coordination network. A hybrid agent capability and associated abstract protocol would be provided for ACL messaging where required.

4.10 Agent Community Augmentation

The cybernetic model lends itself to community augmentation through specialist-agent addition, either via long-distance communications and interoperation or through physical migration. In each case, the central managing body provides the administrative function including specialist-agent capability identification protocols, ontology and language supports. A physical migration instance arises in the case where a new community function cannot be spawned on-board. Distributed interoperation may not be a viable solution if the function needs to be real-time but the spacecraft is out of communication range at that time. The specialist would migrate during ground contact prior to the loss of telecommunications line of sight.

5. Example: An Autonomous Spacecraft Subsystem

Figure 7 illustrates some beginning thoughts on how to depict the infrastructure of an autonomous spacecraft subsystem in the context of the cybernetic (viable system) modeling technique. The "IA" in the figure refers to "Intelligent Agent".

- 1 – An Individual Subsystem.
- [1,2] – Subsystem/Monitor interface.
- [1, 2, IA3] – 1st Tier: showing one member of the control loop set in which each member maintains a spacecraft subsystem.
- IA3 – 1st Tier: Agent that handles a subsystem and its components.
- IA5 – 2nd Tier: Agent that plans, reasons, models, executes, and performs community oriented functions (i.e. spacecraft level).
- 2 - 1st Tier agent perceptor/effector that passes events to IA3 for decisions and receives acts scheduled for transmission.
- 4 - 2nd Tier: Monitor/Governor that reacts to community state changes by monitoring all [2,2] interactions.

- [IA5, 1] – Direct probe of a suspicious subsystem state by Tier 2. Likewise, IA3 probe of one of its subsystem’s components.
- [IA3, IA5] – Agent Messaging Interface. e.g. resolving issues.

Note: “IA3” is the first level of recursion of “IA5”.

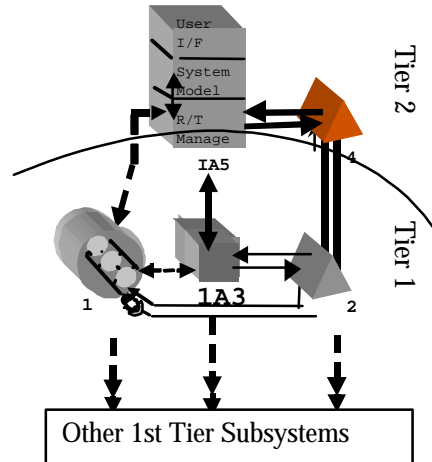


Figure 7. Top-level View of an Autonomous Spacecraft Subsystem

This model begins to show, at a high level, the applicability of the modeling technique in the context of spacecraft subsystems. The model would be duplicated for each subsystem and then integrated into an overall model of the spacecraft. The modeling technique may also be applicable to autonomous science-instruments (this is a new area of interest for us).

6. Conclusion

The task of applying a cybernetic modeling technique to the modeling of an agent community is in its initial stages. This paper is a progress report on our attempts. We feel that this modeling technique will provide a rich representational insight into the various levels of behaviors that will be required in order that an agent community functions as a viable system. So far the technique has been descriptive and seems to address most of the agent-community requirements identified in the earlier portion of this paper. We hope that, as the model becomes more comprehensive in community-behavioral detail, the model can take on a prescriptive role and is of major assistance in the actual development of the agent community being modeled. Time will tell.

References

- [1] Agent Technology from a NASA Perspective, Walt Truskowski et. al., CIA'99 Workshop, Uppsala Sweden, LNCS 1652, Springer, 1999

- [2] Developing Intelligent Agents for Distributed Systems, M. Knapik, J. Johnson, McGraw-Hill, 1998
- [3] Multiagent Systems, J. Ferber, Addison Wesley Longman, English Edition, 1999
- [4] Decision and Control, Stafford Beer, Wiley, 1966
- [5] Heart of the Firm, Stafford Beer, Wiley, 1979
- [6] Brain of the Firm (Second Edition), Stafford Beer, Wiley, 1981
- [7] Beyond Dispute, Stafford Beer, Wiley, 1994
- [8] The Viable System Model, R. Espejo, R. Harndin editors, Wiley, 1989
- [9] Dealing With Complexity, R. Flood, E. Carson, Plenum Press, 1988